20th international Conference on Pattern Recognition of IEEE (August-2010)

**A combined self-configuring method for object tracking in colour video**

Juan Rosell-Ortega
Gabriela Andreu-García
Angel Rodas-Jorda
Vicente Atienza-Vanacloig
Computer Vision Group. DISCA. UPV jarosell@ai2.upv.es, {gandreu, arodas, vatienza }@disca.upv.es

Abstract
This paper introduces a novel approach to back- ground modeling. We propose using initially a method to extract scene parameters from a sequence of frames. These parameters, together with an initial background model, are used as a starting point for a background subtraction method based on fuzzy logic. Our method permits modeling the background and detecting moving objects in a video sequence without user intervention. The algorithm is designed to work with CIEL*a*b* coordinates with multi modal support and eludes user parameters or fixed or probabilistic thresh- olds usually found in the traditional background sub- traction methods. Quantitative and qualitative results obtained with a well-known benchmark and comparisons with other approaches justify the model.

**Acknowledgments**
This work has been partially supported by SENSE project (Specific Targeted Research Project within the Thematic priority IST 2.5.3 of the 6th Framework Program of the European Commission: IST Project 033279), and by the Spanish Government by the complementary funding TIN2007-30367-E

SENSE GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN

2010 International Conference on Pattern Recognition

# A combined self-configuring method for object tracking in colour video

Juan Rosell-Ortega*, Gabriela García-Andreu†, Ángel Rodas-Jordà†, Vicente Atienza-Vanacloig†
*Instituto de Automática e Informática Industrial (AI2) - Universidad Politécnica de Valencia
Valencia, Spain
Email: jarosell@ai2.upv.es
†Dpto. De Sistemas y Computadores - Universidad Politécnica de Valencia
Valencia, Spain
{gandreu, arodas, vatienza }@disca.upv.es

***Abstract*—This paper introduces a novel approach to background modelling. We propose using initially a method to extract scene parameters from a sequence of frames. These parameters, together with an initial background model, are used as a starting point for a background subtraction method based on fuzzy logic. Our method permits modelling the background and detecting moving objects in a video sequence without user intervention. The algorithm is designed to work with CIEL*a*b* coordinates with multi modal support and eludes user parameters or fixed or probabilistic thresholds usually found in the traditional background subtraction methods. Quantitative and qualitative results obtained with a well-known benchmark and comparisons with other approaches justify the model.**

## I. Introduction

Background subtraction [1] is one of the most popular methods to detect regions of interest in frames. This method consists in finding the differences of incoming frames with respect to a background model in order to detect pixels whose difference is over a threshold.

A popular background subtraction method consists in modelling each pixel with a Gaussian distribution. Regions of interest are detected by subtracting each new frame from the average image and thresholding the result. The adaptive version of this algorithm updates the model parameters recursively by using a filter [2]. This model does not work well when the background is not static. Authors of [3] propose using more than one Gaussian to solve this issue. In [4] it is proposed to build an statistical representation of the background by estimating directly from data the probability density function. In [5], authors propose a method in three stages. At the lowest level, a Wiener filter is used to classify pixels. In another stage, foreground pixels are recovered with the information provided by previous segmentations. Finally, different models are kept to face model corruption. Authors of [6] propose using an exponential to compute background and foreground probability, pondering the distance to the background and the motion of a pixel. Radically different is the approach introduced in [7], in which it is proposed using LBP histograms to model the background. By using histogram intersection, differences with background are detected. Other approaches proposed in recent years are computing a histogram of edges in a block basis, found in [8] or the use of salient motion introduced in [9]. Few papers propose the use of fuzzy logic in computer vision applications, being most of the work related to clustering techniques. In [10] an approach to using fuzzy logic in segmentation problems is described.

The goal of this paper is discussing a technique that allows the system to obtain the parameters it needs to describe the scene, including an initial background model $B(0)$, from a sequence of CIEL*a*b* frames $F(0), F(1), ...F(n)$ by using BAC [6]. Once the parameters are obtained, another, more accurate and robust, algorithm based on fuzzy logic, is executed to proceed with background subtraction.

## II. Fuzzy background subtraction

We use a background membership function to approximate the threshold that separates background and foreground after each background subtraction.

Following the approach of [5], our algorithm works at two different stages: pixel and region. First, only pixel operations are performed, such as background subtraction and model update. In the region level,the goal is recovering foreground

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.1154

pixels lost due to an inaccurate segmentation, by considering former segmentations. For each frame $F(i)$, the background subtraction is computed as the euclidean distance $d^v_{x,y}$ of the CIEL*a*b* coordinates of each pixel $F_{x,y}(i)$ from each $B^v_{x,y}(i)$, being $B^v_{x,y}$ the $v$-th model for pixel in $x, y$ in time $i$. The distance finally considered is,

$$d_{x,y} = min(d^1_{x,y}, d^2_{x,y} ... d^v_{x,y}), \forall v \tag{1}$$

Figure 1 shows the result of informal experiments with different images. It can be seen that background pixels cluster close to distance zero, as expected. This fact may be translated into a membership function as shown in figure 2. In it, $d$ is the distance between two pixels and the $Y$-axis represents the membership value. Figure 2 shows two functions, on the left, the background membership function ($mB$) and on the right, its inverse, the foreground membership function ($mF$). Up to a distance $a$, pixels have a background membership value of 1. Background membership decreases as distance increases. We used a straight line to model this decrease. Using $d_{x,y}$, the algorithm computes $mB(x, y)$ and $mF(x, y)$, the membership values of $(x, y)$ depending on $d_{x,y}$. First, the process computes the distance $d_0$ that gathers the first $q_{min}$ pixels in the range of distances from 0 to $d_0$ and set $mB(d) = 1, \forall\ 0 \leq d \leq d_0$, the interval at which the background membership function is evaluated to 1. The algorithm computes the cardinality of the pixels at a distance $d_i$ as,

$$Card(d_i) = Card(\{F_{x,y}/d_{x,y} \leq d_i\} \tag{2}$$

Iteratively, it looks for the distance at which small increments of elements are detected. This is done by computing the slope of the straight

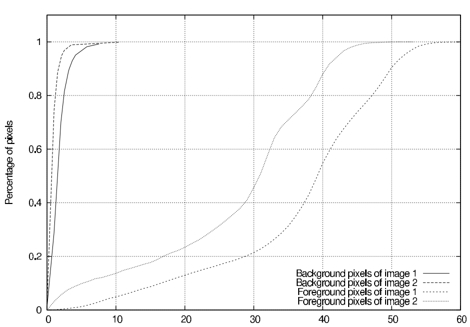


Fig. 1: Plot of background and foreground pixels cumulated by distance. Lines on the left of the plot represent the percentage of background pixels of each frame. Lines on the right represent the percentage of foreground pixels.

line between distances $d_i$ and distance $d_{i+1}$ as $m_i = \frac{Card_(d_i) - Card(d_{i+1})}{d_{i+1} - d_i})$. Distance $d_1$ is the third consecutive distance with an inclination angle $\beta < \beta_0$, to avoid false detections due to noise. Being $m$ the slope between $d_0$ and $d_1$, the membership function then can be written as,

$$mB(d) = \begin{cases} 1,\ if\ 0 \leq d \leq d_0 \\ m \times d + b,\ d_0 < d \leq \frac{-b}{m} \\ 0,\ \frac{-b}{m} < d \end{cases} \tag{3}$$

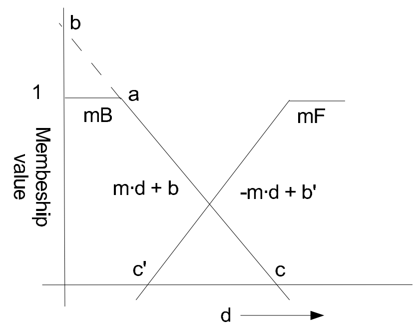


Fig. 2: Representation of two fuzzy membership functions $mB$ and $mF$. Note the symmetry of both functions.

## III. Background modelling

Multi-modal support background modelling algorithms ( [3], [7], [5]...) need two parameters set by user $K$, which limits the maximum number of models allowed per pixel and $\gamma$, a factor that controls the frequency in which models are modified. In this section we discuss an approach to extract them from a reduced sample of frames from the scene. BAC (Background Algorithm with Confidence) computes the confidence of a pixel in such a way, that it reaches easily values around 0.9, over a maximum of 1, after 10 or 20 consecutive frames without changes. If we apply this algorithm to the input sequence until the computed background model achieves a certain confidence $\theta$, we obtain an initial model $B_{x,y}(0)$ and for each pixel $(x, y)$, $h(x, y)$, the number of times the model was modified, and $g(x, y)$, the confidence obtained. It is easy to see that the values of $K$ and $\gamma$ have an strong relationship with the values of $h$ and $g$. The bigger the value of $h(x, y)$, the most models pixel $(x, y)$ may need, and the higher the confidence $g(x, y)$ obtained, the lowest its need of changing quickly. As models should show flexibility to changes, $K$ may be estimated following the rule:

TABLE I: Results (in %) of the compared algorithms when faced to the different situations represented in the Wallflower benchmark. Row lalbeled as $TP$ stands for true positives, and $TN$ stands for true negatives.

| *Method* | *Seq.* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
|---|---|---|---|---|---|---|---|---|
| | TP % | 44 | 50 | 92 | 73 | - | 41 | 86 |
| *Mix. Gauss.* | TN % | 95 | 85 | 73 | 07 | 100 | 98 | 90 |
| | TP % | 71 | 62 | 73 | 79 | - | 86 | 85 |
| *LBP* | TN % | 90 | 91 | 86 | 66 | 100 | 97 | 77 |
| | TP% | 60 | 75 | 56 | 48 | - | 30 | 83 |
| *BAC* | TN% | 90 | 76 | 87 | 90 | 100 | 98 | 67 |
| | TP% | 77 | 97 | 99 | 70 | - | 79 | 85 |
| *Wallflower* | TN% | 99 | 98 | 90 | 98 | 100 | 99 | 85 |
| | TP% | 61 | 68 | 80 | 68 | - | 79 | 94 |
| *Ours* | TN% | 95 | 93 | 91 | 85 | 99 | 98 | 89 |

TABLE II: Space complexity for the algorithms, taking an image of $n \times m$ pixels.

| *Algorithm* | weight | model |
|---|---|---|
| *Mix. Gauss.* | $O(n \times m \times K)$ | $O(n \times m \times K)$ |
| *LBP* | $O(n \times m \times K)$ | $O(n \times m \times K \times 2^P)$ |
| *Wallflower* | - | $O(n \times m \times V)$ |
| *BAC* | - | $O(n \times m)$ |
| *Ours* | $O(n \times m \times K)$ | $O(n \times m \times K)$ |

$$\forall(x,y),\ K(x,y) = \begin{cases} 2,\ \text{if}\, h(x,y) \leq 2,\ \forall(x,y) \\ min(5, h(x,y)),\ \text{if}\, h(x,y) > 2, \end{cases} \quad (4)$$

and $\gamma(x,y) = 1 - g(x,y)$, $\gamma(x,y)$ is restricted to the interval $[0.3, 0.6]$, in order to avoid that a pixel never changes or never arrives to an stable state. Model $B_{x,y}(0)$ is set to the model obtained by BAC. Background pixels are those which meet the condition $mB(d_{x,y}) > 0.5$ and are labelled in a binary image $S(x,y) = 0$, for foreground pixels $S(x,y) = 1$. The background model is updated depending on information of $S(i)$. If $S_{x,y} = 0$ and $d^v_{x,y} = min(d^v_{x,y})$, then model $v$ matches the input and its colour and confidence are updated as follows,

$$B^v_{x,y}(i) = \alpha B^v_{x,y}(i-1) + (1-\alpha) F_{x,y}(i) \quad (5)$$

$$c^v_{x,y}(i) = \alpha c^v_{x,y}(i-1) + (1-\alpha) mB(d^v_{x,y}) \quad (6)$$

The non matched models for pixel $(x,y)$ update their confidences as $c^v_{x,y}(i) = \alpha\, c^v_{x,y}(i-1)$, where $\alpha$ is a learning rate factor in $[0,1]$ and $B^v_{x,y}(i-1)$ is the value in time $i-1$ of $v$-th model of pixel at $(x,y)$. Models of $(x,y)$ are ordered in descending order according to their confidences. If the condition $\gamma > \sum_{v=1}^{K} c^v_{x,y}(i)$ is verified, then a new model will be added or the $K$-th model be replaced as $B^K_{x,y}(i) = F_{x,y}(i)$ and $c^K_{x,y}(i) = 0.01$. In [5], finding an unusual percentage of foreground pixels is enough to change the background model.

Our approach detects model corruption by testing if $d_0 > d_{corrupt}$ . The process restarts and keeps only one model per pixel. It sets $B^1_{x,y}(0) = F_{x,y}(i)$, $c^1_{x,y} = max(c^v_{x,y})$. Restarting the process does not mean starting again with BAC, as the model's parameters should not be changed. At region level, we followed a similar approach as in [5], using a colour list for each region instead of a gray tone histogram. The intersection of adjacent pairs of differenced images and the previous foreground image is computed $P = S'(i) \wedge S'(i-1) \wedge S(i-1)$. Connected regions $R_i$ of $P$ with less than $p_{min}$ pixels are discarded. Colours of each $R_i$ are stored in a set $C_i$ considering as equal colours with a difference under $c_{min}$ and for each $(x,y) \in R_i$ in frame $F(i)$ a binary image is computed,

$$L_{x,y} = \begin{cases} 1,\ if\ \exists c \in C_i \mid dist(c, F_{x,y}(i)) \leq c_{min} \\ 0,\ otherwise \end{cases} \quad (7)$$

being $dist(c, F_x(i))$ the euclidean distance of two colours. $S(i)$ is grown by performing,

$$S(i) = S(i) \vee L \quad (8)$$

## IV. Experiments

Experiments were performed with the Wallflower benchmark [5], which seems to be accepted as a common validity test. We compared four algorithms to ours: the Wallflower algorithm [5], the LBP algorithm [7], BAC [6] and Stauffer's approach [3]. Qualitative results are shown in figure 3. For Stauffer's algorithm we used $K = 5$ and $T = 0.6$. The others were executed with the parameters as stated by their authors.

For fuzzy background subtraction, we took $q_{min} = 30\%$ of pixels, $\beta_0 = 1°$, $d_{corrupt} = 10$, $p_{min} = 8$, $c_{min} = 7$, $\theta = 0.9$. In all cases, $\alpha = 0.99$. Table I shows quantitative results, we measured the foreground and background pixels classified successfully, $TP$ and $TN$. BAC has in average the worst results, because its aim is providing another algorithm with an initial background model and useful statistics of the scene. Only Wallflower and our approach succeed in sequence *lightSwitch* due to the region level processing. The use of the colour bank result in some foreground pixels lost in *foregroundAperture*, but yielding better results than others. Table II compares Algorithms' space complexity.


Fig. 3: Results for the Wallflower benchmark. On the left, original control frames captured from the sequence. The second column from the left represents the control frames segmented by hand. Remaining columns represent the result of each algorithm for each sequence.

## V. Conclusions [1]

We introduced a novel approach for background modelling using the combination of an algorithm to extract estimations used as parameters for another, more accurate, background modelling algorithm using fuzzy logic. The detection of model corruption is defined according to the membership functions computed. Experiments show that ours yields qualitative and quantitative results comparable to other algorithms with the lowest space complexity.

## References

[1] M. Piccardi, "Background subtraction techniques: a review," *IEEE Transactions on Systems, Man and Cybernetics.*, vol. 4, pp. 3099–3104, Oct 2004.

[2] C. . R. Wren, T. D. A. Azarbayenjani, and A.P.Pentland, "Pfinder: rel-time tracking of the human body," *IEEE Trans. PAMI*, no. 7, pp. 780 – 785, 1997.

[3] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE CVPR99*, pp. 246 – 252, 1999.

[4] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *ECCV 2000*, pp. 751 – 767, 2000.

[1] Supported partially by the research projects DPI2007-511 66596-C02-01 (VISTAC) and project 033279 (SENSE)

[5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," *IEEE ICCV99*, pp. 255–261, 1999. [Online]. Available: http://research.microsoft.com/ jckrumm/wallflower/testimages.htm

[6] J. Rosell-Ortega, G. Andreu-García, A. Rodas-Jordá, and V. Atienza-Vanacloig, "Background modelling in demanding situations with confidence measure," *IEEE ICPR08.*, 2008.

[7] M. M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. PAMI06*, vol. 28, no. 4, pp. 657–662, April 2006.

[8] M. Mason and Z. Duric, "Using histograms to detect and track objects in color video," *Proc. Applied Imaginery Pattern Recognition Workshop*, pp. 154– 159, 2001.

[9] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. PAMI00*, no. 8, pp. 774 – 780, 2000.

[10] C. Zhang and P. Wang, "A new method of color image segmentation based on intensity and hue clustering," *IEEE ICPR00*, vol. 3, 2000.